GenAI Engineer Candidate Task: Intelligent Document Understanding & AI-Ready API Design

Time Allotment: Around 3-4 hours

Objective

This task assesses your ability to design and implement AI-powered document understanding systems and create AI-ready APIs. You'll build a foundational system that aligns with Factify's vision of transforming static documents into intelligent, API-accessible assets.

Scenario

Factify is building a next-generation document platform where documents have intelligent metadata accessible through APIs. We need to develop an AI system that can understand documents and make their metadata available through a well-designed API.

Task

Part 1: Document Intelligence System

Build a system that processes 3 provided business documents and extracts intelligent metadata using AI.

Provided Files:

- `invoice.pdf` - A typical business invoice (we have 2 files)
- `contract.pdf` - A legal contract with multiple clauses
- `earnings.pdf` - A quarterly business report with tables and charts

Required Extractions:

1. **Document Type Classification**
    - Use zero-shot or few-shot classification with LLMs
    - Explain your prompt engineering approach
    - Provide confidence scores

2. **Semantic Metadata Extraction**
    - Design a flexible extraction system that adapts to document type

    - Extract key information based on document type:
        - Invoice: vendor, amount, due date (if applicable), line items
        - Contract: parties, effective date, termination date, key terms
        - Report: reporting period, key metrics, executive summary
    - Handle cases where expected fields are missing gracefully

Part 2: AI-Ready API Design

Design and implement an API that makes documents and their metadata accessible in an "AI-friendly" way.

Required Endpoints:

1. `POST /documents/analyze`
    - Accepts a document for processing
    - Returns the extracted metadata
    - Include processing status and confidence scores
2. `GET /documents/{id}`
    - Returns a document with all its metadata
    - Structure the response to be easily consumable by AI agents
    - Include semantic descriptions of fields
3. `GET /documents/{id}/actions` - This one should be a mock so use static data in a format you decide on
    - Returns all actionable items for a specific document
    - Include filtering by status, deadline, or priority

API Design Requirements:

- Use clear, semantic field names
- Include field descriptions in responses (AI-friendly)
- Design consistent error responses
- Provide example responses for each endpoint

Part 3: Talking Points (Written Part)

1. Explain your design decisions
2. **Propose 2 AI-Powered Features** for Factify:
    - Feature 1: Using the metadata you extract, propose an intelligent feature
    - Feature 2: How could AI make document workflows smarter?
    - For each: explain the technical approach and business value
3. **Production Considerations**:
    - How would you handle LLM API failures?
    - Propose a simple caching strategy to reduce API calls
    - Estimate the cost per document processed

Technical Requirements

Implementation Guidelines:

- Use Python
- Use at least one LLM API (OpenAI, Anthropic, or open-source)
- It is allowed and recommended to work with coding assistance tools (i.e. cursor/Claude/Windsurf etc.)
- Feel free to to find additional documents to help you validate your work (notice that we would test it on additional ones)

Output Format:

JSON Output for each document **(Just a suggestion, feel free to use something else, as long as it is documented)**:

json

```
{
  "document_id": "generated_uuid",
  "filename": "invoice_sample.pdf",
  "classification": {
    "type": "invoice",
    "confidence": 0.95
  },
  "metadata": {
    // Type-specific fields based on your extraction
  }

}
```

Submission

Create a GitHub repository with:

- Document processing script
- API implementation
- `output/` folder with JSON results for all 3 documents
- `api_docs.md` with endpoint documentation **Please give an example of running each of the endpoints!**
- README with:
  - Setup instructions
  - Explanation of your approach
  - Part 3 of the exercise

Good luck!